RECOGNITION AND VERIFICATION OF TOUCHING HANDWRITTEN NUMERALS

Zhou, J.; Kryzak, A.; Suen, C.Y.

Recognition and Verification of Touching Handwritten Numerals

Jie Zhou
IBM Toronto Laboratory
330 University Avenue, Toronto Canada M5G 1R7
E-mail: jiez@ca.ibm.com

Adam Krzyzak, Ching Y. Suen
Centre for Pattern Recognition and Machine Intelligence
Concordia University, Montreal, Canada H3G 1M8
E-mail: {krzyzak,suen}@cenparmi.concordia.ca

In the field of financial document processing, recognition of touching handwritten numerals has been limited by lack of good benchmarking databases and low reliability of algorithms. This paper addresses the efforts toward solving the two problems. Two databases IRIS-Bell'98 and TNIST are built/organized to serve as standard data sets. Working with the samples from these databases, we proposed a Recognition & Verification system measured by precision rate, which reflects the system reliability in a class-specific manner. The graph-based recognizer combines the segmentation-based and segmentation-free approaches, while the verifier incorporates both general and domain specific verification schemes. Results supported the effectiveness of the proposed verification scheme.

1 Introduction

In today's information society, businesses and organizations must process immense numbers of business and financial documents that include handwritten numerals, such as bank cheques, payment slips, and tax forms. Recent advances in recognition technology have allowed the development of automatic processing systems[1,2]. However, many problems are still hindering the emergence of completely successful systems. A particular problem is the recognition of touching numerals, most commonly touching numerals pairs. This problem has been gaining the attention of researchers due to its important practical role [3,4,5], but less work has been done here than on the recognition of isolated numerals. There are two main obstacles to this research:

1. There is a lack of benchmarking data to support research efforts. CEDAR CD-ROM database does contain some images of touching numerals, provided as mixture with isolated numerals. However, it contains only a few hundred entries. These entries have been collected from zip codes, and thus do not represent the features of touching numerals typically encountered in financial documents. There is a critical need for databases that

In: L.R.B. Schomaker and L.G. Vuurpijl (Eds.), Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, September 11-13 2000, Amsterdam, ISBN 90-76942-01-3, Nijmegen: International Unipen Foundation, pp 179-188

are standardized for financial domains, as well as databases of sufficient sizes for training and testing.

2. The reliability of the existing algorithms for touching numerals is low. For the practical systems used in financial environments such as banks, revenue departments or other financial institutions, there is a gap between the current and expected system reliability.

This paper describes two databases for touching numerals, and presents a verification scheme that aims to increase the reliability. We view the classification as a two-stage process: indexing and verification. Indexing is to label a class identity on an input sample using a general-purpose recognizer, while verification is to confirm or negate that label in a reliable way. Section 2 describes the construction of two databases, that have been designed to serve as standard training and testing sets for touching numeral recognition. Section 3 briefly introduces the segmentation method and general-purpose recognizer for the touching numerals. Section 4 focuses on verification scheme. Concepts and methodologies are proposed, a Recognition & Verification system for touching numerals are presented with results analysis. Section 5 gives the conclusion.

2 Databases

Two databases for touching numerals are built and intended to serve as standard data sets for other researchers in the field. One is IRIS-Bell'98 collected by CENPARMI researchers from financial domains. The other is TNIST database extracted from NIST CD ROM SD 19.

IRIS-Bell'98 consists of two components: IRIS-Cheque and Bell'98. The IRIS-cheque includes samples of Canadian personal cheques written by employees/students of Concordia University and employees of Bell Canada. Bell'98 includes samples of phone bills written by the general public. Consequently, all the data in IRIS-Bell'98 have the background of real-life financial document. The document is first scanned with a resolution of 300 dots per inch, then the area of courtesy amount is extracted. Isolated and touching numerals are distinguished automatically or manually. IRIS-Bell'98 contains a training set of 2538 touching numeral pairs and a test set of 1193 pairs. One obvious feature of this database is its very free style. Although the size is not very big, the database represents some difficulties of real situation for financial document processing system.

TNIST database is built upon NIST SD19. The SD19 contains well-organized isolated numeral and alphabetic characters that have been widely used as standards in the community; however, some extra efforts are needed to produce standard datasets for touching numerals. We built our TNIST in two steps: first extracting touching numerals from the form images of SD19, then tagging these touching numerals with their identities and organizing them as a database. The database is separated into training and testing samples according to the original partition of NIST SD 19. Consequently, the database contains a training set of 4252 touching numeral pairs and a test set of 4395 touching numeral pairs.

3 General-purpose Recognizer

The recognition of touching numerals can be categorized as segmentation-based, segmentation-free or holistic. We developed a graph-based scheme called Segmentation Recognition Cost Graph (SRCG) in our system. It combines segmentation module and dual-recognizer into an overall consideration.

The segmentation algorithm as proposed in [6] solves slanting and some overlapping problems by contour tracing in the segmentation stage. After segmentation, we get a list of possible cuts in a sorted sequence. A filtering can be done to decrease the number of possible cuts that are passed to SRCG.

Figure 1 illustrates the use of the SRCG scheme to recognize a touching digit pair. Results and confidence values of the segmentation module and recognizers are put on the arches of the graph. The path from "start" to "end" with the highest combined value (lowest cost) is selected to give the best result.

The two recognizers used in SRCG are the neural approaches described in [2] and [7] respectively. The scores for these recognizers are the confidence values of the output. In the illustrative example of Figure 1, a touching "59" goes through the segmentation module. After segmentation candidate filtering, we got two possible candidate cuts. We feed the left and right subimage of each possible segmentation into two recognizers and get four decisions with the confidence values on the graph edges. Only lines with the same style (solid or dotted) can be a possible combination. We can obtain 8 possible paths. The highest is the conclusion of '5' and '9' with a total sum of 3.0.

The graph rejects a decision when both of the following conditions are satisfied: (1) two top scores with different conclusions are smaller than a threshold; (2) for the best path, the difference between two recognizers gives different conclusions.

Experiments are conducted based on IRIS-Bell'98 and TNIST databases for touching numeral pairs. The results of the general-purpose recognizer are listed in Table 1. Performances are also computed according to the measurement of Precision Rate, which reflects system reliability in a class-specific

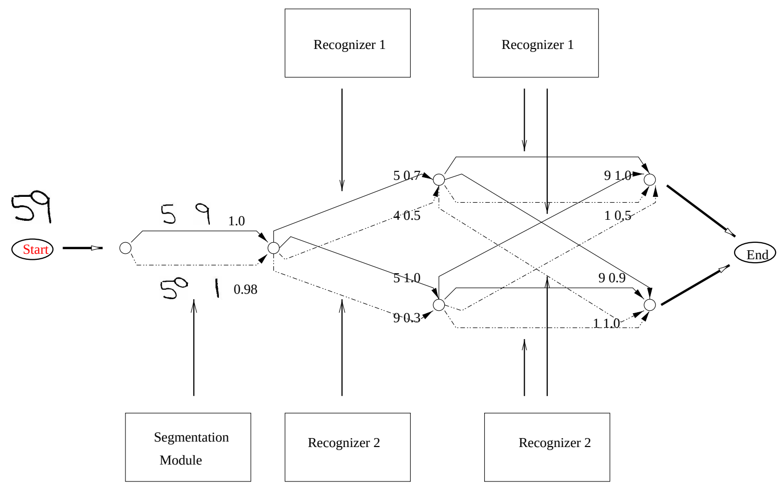

Figure 1: Segmentation Recognition Cost Graph

manner and accurately represent the characteristics of verifiers. The definition is provided in the Appendix. The comparison of precision rate will be used to illustrate the effectiveness of verification scheme in next section.

Database	Correct	Rejection	Precision for Strings	Precision for Numerals
IRIS-Bell'98	65.5%	16.3%	78.5%	84.3%
TNIST	88.6%	3.1%	90.1%	94.2%

Table 1: Test Results of General-purpose Recognizer (without verifier)

4 Verifier

The Recognition & Verification system embeds a pattern verifier after the general-purpose recognizer in the conventional system:

Pattern Verifier: A specially trained expert which confirms or negates a preset pattern class, usually used after a general-purpose recognizer, with the intention to significantly improve class-specific precision rates of the system.

In our research on isolated handwritten numerals, a novel kind of neural network, Quantum Neural Network (QNN), has been shown to be a good candidate for verification purpose due to its inherent fuzzy feature. Its ability to encode uncertainty along the decision boundary makes it good at distinguishing confusing numerals. With QNN as building blocks, the verifier for isolated numerals achieved satisfactory precision rates. Details can be found in [8] and [9].

When the problem comes to touching numerals, since we are not using holistic approach, QNN can still be applied in the verifier. However, the scheme is not enough. The reason is that the general-purpose recognizer suffers from specific weaknesses arising from the domain of touching numerals. In this section, we will first describe the domain specific problems and solutions, then present the complete diagram of the verifier for touching numerals, which makes use of both QNN and the domain specific solutions.

4.1 Domain specific verification

Two domain specific verification schemes are applied:

1. Wrong segmentations which can be detected by Touching Type & Location verification.

   In discrete approach of recognizing touching numerals, the proper segmentation is very important. If a wrong segmentation occurs, even a perfect recognizer will not help. Our general-purpose recognizer, which considers multiple splitting possibilities with recognizer assistance in an efficient way, partially relieves the headache of pure segmentation-based algorithm. However, it is far from worry-free. In the example presented in Figure 2, due to the smooth connection of the loop in '6' and the stroke in '5', none of the three segmentation paths separates the left '6' from the '5'. To solve this problem, we propose Touching Type & Location Verifier. It is based on the possible touching types between two numerals and location of the touching zone. Given a segmented numeral, if its touching location and touching type are impossible to occur together, the decision of general-purpose recognizer is negated.

   Touching types can be categorized according to the stroke type we get after segmentation. If the cutting points create an end of a stroke, we call it an End Point (EP). Otherwise, it is a Non-end Point (NEP), which can be the side of a stroke or a bend area. We define 4 touching types for a touching numeral pair, which will be used in Touching Type & Location Verification: (1) EP-NEP or NEP-EP; (2) EP-EP; (3) NEP-NEP; (4) MULTIPLE. Touching location is defined for a segmented numeral from touching pairs. It is divided into two kinds: upper-touching and lower-touching.

   Take Figure 2 as the illustrative example. For real numeral '1', the touching with another numeral usually occurs along the side of the stroke,

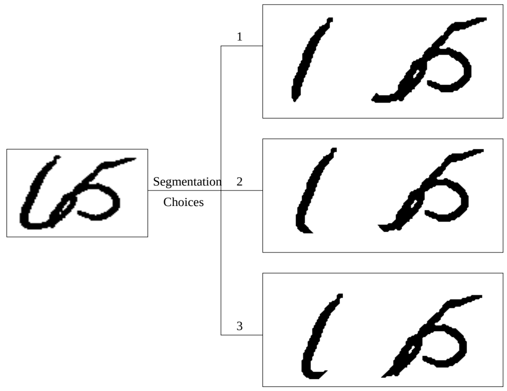

Figure 2: Problematic Segmentation Solved by Verifier

i.e., it is a NEP type touching. If a segmentation creates an end point in the lower part, it is highly possible that one numeral is mistakenly cut into two and leave a '1' like segment as the left image.

2. Errors caused by ligatures and/or weakness of recognizers which can be solved by the structural verifier making use of structural features.

   Touching numerals tend to have more structural variations compared with isolated numerals. These variations can be introduced by ligatures, overlaps and some degree of cursive styles. The situation can be worsened by inappropriate segmentation.

   Figure 3 gives an example when cursive style and the ligature cause some distortion of the numeral '0' in pair "80". Examination of the first segmentation possibility reveals that the recognizers caught the global shape of the right hand image which is similar to '7'. Although the confidence value is not high (0.6), it is higher than the confidence value of '0' in the second segmentation possibility (0.3). It gives a final conclusion of "87". The problem is not as fatal as errors of segmentation and it is likely to be negated by the QNN verifier. However, we can see that despite distortion, the basic structural feature of numeral "0" — the big loop — persists in the image, which can be used for verification purpose.

   In our Recognition & Verification system, we decide to put structural features in verification stage instead of general-purpose recognizer due to the following considerations:

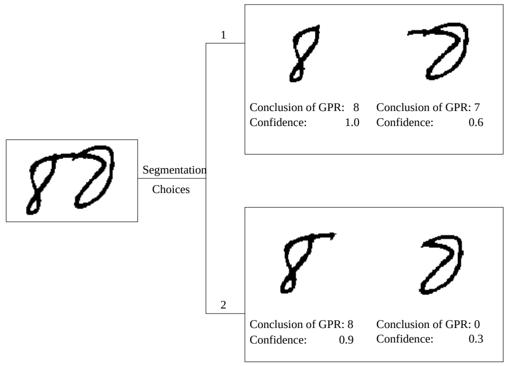

Figure 3: Structural Distortion Solved by the Verifier.

- Structural features are intuitive and explicit observations which match human expect verification angle. In a practical system, if the user can not understand the cause of the remaining errors from her/his point of view, the system tends to be rejected. Thus a highlight of the verifier is the needs of the real users.
- Structural verifiers are class-specific. They have a pre-determined target, which makes the scheme computationally effective. Structural features are very effective in indicating some dominant features for confirmation or negation as verification purpose. However, it would be a painstaking process to develop complete sets of rules for a ten-class general-purpose recognizer.

4.2 Diagram of verifier

We propose the complete architecture of verifier for touching numerals depicted in Figure 4.

The verifier has a hierarchical architecture. When an identity conclusion of a segmented image comes from the general-purpose recognizer, it first goes through dummy symbol checking. If it is considered as a dummy by a dummy symbol detector, further verification dedicated for numeral classes becomes unnecessary. Otherwise, class specific Touching Type & Location Verification and Structural Verification are applied. If the conclusion is negated by either schemes, the system will go back to the next segmentation candidate of different conclusion for a second chance. If both schemes accept the conclusion, the candidate goes straight to the QNN Verifier for a final verification. The result

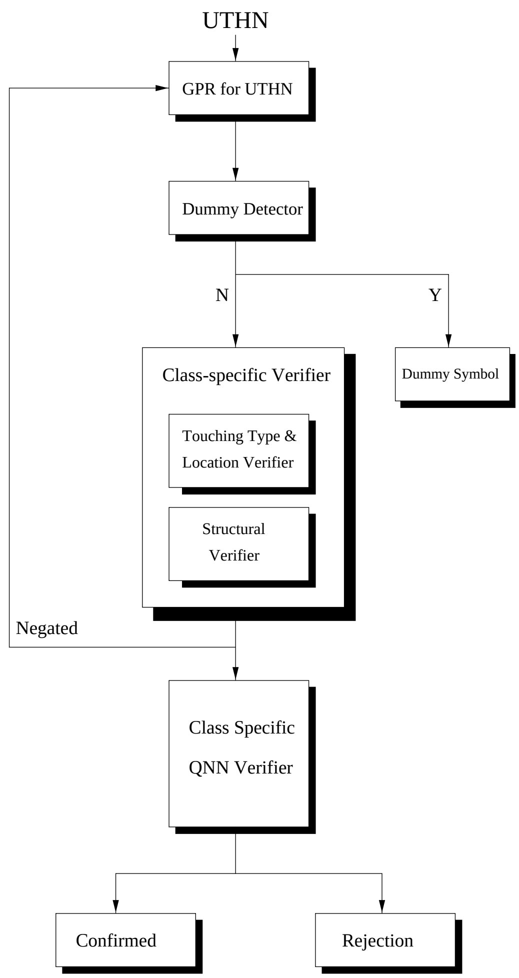

Figure 4: Diagram of Verifier in R&V system. UTHN stands for Unconstrained Touching Handwritten Numerals.

will be either a confirmation or a rejection.

The experiments were conducted on IRIS-Bell'98 database and TNIST database. The complete R&V system produced the performance shown in Table 2.

Compared with the results from general-purpose recognizer, the precision improvement is obvious, while the recognition rate does not drop much. It can be explained by the reliable negation scheme with a low false negative rate, and the helpful second chance loop.

The reliabilities of our system, reflected by precision rates, is comparable or better than those reported in literature [4,5,3]. However, lacking of a well known standard set limits the possibility of further comparison. We expect that the building of IRIS-Bell'98 and TNIST databases will provide an open and standard base for researchers to exploit and compare techniques of the problem.

Database	Correct(%)	Substitution(%)	Precision for Strings(%)
IRIS-Bell'98	65.2	9.2	89.2
TNIST	85.7	3.5	96.1

Table 2: Test Results of R&V System

5 Conclusion

This paper describes two data sets built for touching numeral recognition as well as a Recognition & Verification system designed to improve the system reliability. A graph-based scheme is used as the recognizer which combines segmentation module and dual-recognizer into an overall consideration. The verifier considers verification scheme for isolated numerals as well as the schemes specific for touching numerals. The performance enhancement measured by precision rate proves the effectiveness of the verifier in terms of improving system reliability.

Appendix

Precision Rate (PR): The precision rate of class C_m is the measurement of how well ONLY the input patterns of class C_m are correctly classified.

Consider subset S_m of the input patterns as the set of samples belonging to C_m. Let K_m be the number of input patterns classified as class C_m by the system. Within K_m, there are X_m patterns *truly* belonging to S_m. The precision rate of C_m is then given by

$$PR_{C_m} = X_m / K_m \tag{1}$$

The precision rate of the system is given by

$$PR_{sys} = \frac{1}{n_0} \sum_{m=1}^{n_0} PR_{C_m} \tag{2}$$

where n_0 is the total number of classes.

References

1. G. Dimauro, S. Impedovo, G. Pirlo, and A. Salzo. Handwriting recognition: State of the art and future trends. In Nabeel A. Murshed and Flavio Bortolozzi, editors, *Advances in Document Image Analysis*, pages 1–18. Springer, Heidelberg, Germany, Nov 1997.
2. C. Y. Suen, L. Lam, D. Guillevic, N. W. Strathy, M. Cheriet, J. N. Said, and R. Fan. Bank check processing system. *Intl. Journal of Imaging Systems and Technology*, 7:392–403, 1996.
3. T. M. Ha, M. Zimmermann, and H. Bunke. Off-line handwritten numeral string recognition by combining segmentation-based and segmentation-free methods. *Patt. Recog.*, 31(3):257–272, 1998.
4. Z. Shi and V. Govindaraju. Segmentation and recognition of connected handwritten numeral strings. *Patt. Recog.*, 30(9):1501–1504, 1997.
5. X. Wang, V. Govindaraju, and S. Srihari. Holistic recognition of touching digits. In *Proc. of IWFHR98*, pages 295–303, Taejon, Korea, August 1998.
6. N. W. Strathy, C. Y. Suen, and A. Krzyzak. Segmentation of handwritten digits using contour features. In *Proc. of ICDAR'93*, pages 577–580, 1993.
7. J. Zhou, Q. Gan, and C. Y. Suen. A high performance hand-printed numeral recognition system with verification module. In *Proc. of ICDAR'97*, vol 1, pages 293–297. IEEE Computer Society, 1997.
8. J. Zhou, Q. Gan, A. Krzyzak, and C. Y. Suen. Recognition of handwritten numerals by quantum neural network with fuzzy features. *International Journal on Document Analysis and Recognition*, 2(1):30–36, 1999.
9. Jie Zhou. *Recognition and Verification of Unconstrained Handwritten Numerals*. PhD thesis, Concordia University, 2000.